

# **Управление файловой системой**



## Признаки файла:

- ✓ файл объединяет множество данных;
- ✓ обладает именем;
- ✓ располагается на внешнем устройстве;
- ✓ предполагает многократное использование информации с разрывом во времени;
- ✓ предполагает совместное использование информации несколькими приложениями или пользователями одновременно (разделяемый ресурс) или с разрывом во времени.



**Файловая система** – часть операционной системы, которая организует эффективную работу с данными, хранящимися во внешней памяти, и обеспечивает пользователю удобный интерфейс при работе с такими данными.

**Файловая система** – это подсистема, включающая:

- ✓ наборы структур данных, используемых для управления файлами;
- ✓ комплекс системных программных средств, реализующих управление файлами.



## **Функции файловой системы:**

1. Идентификация файлов.
2. Распределение внешней памяти между файлами.
3. Обеспечение надежности и отказоустойчивости.
4. Обеспечение защиты от несанкционированного доступа.
5. Обеспечение совместного доступа к файлам.
6. Обеспечение высокой производительности.



**Файловая система** — это функциональная часть ОС, обеспечивающая выполнение операций с файлами.

**Файл** — это логически связанная совокупность данных или программ, для размещения которых во внешней памяти выделяется определенная область.



## **Общая задача файловой системы:**

предоставление пользователю логической модели для работы с файлами и отображении этой модели на физическую организацию внешнего устройства.

Логическая модель обеспечивает удобный для пользователя интерфейс и скрывает физическую организацию работы с внешними устройствами.



## Логические операции и средства:

- ✓ именованние файлов;
- ✓ поддержка различных типов файлов;
- ✓ задание атрибутов файлов;
- ✓ организация хранения множества файлов;
- ✓ поддержка логической организации файлов;
- ✓ предоставление программного интерфейса для работы с различными файлами (системные функции, например, WinAPI).



## Физическая организация внешнего устройства предполагает:

- ✓ отображение имен файлов в адреса внешней памяти;
- ✓ размещение данных на устройстве;
- ✓ обеспечение доступа к данным;
- ✓ организация совместного использования файлов;
- ✓ защита файлов одного пользователя от несанкционированного доступа другого;
- ✓ восстановление файлов в случае возникновения ошибок различного рода;
- ✓ обеспечение устойчивости файловой системы к сбоям питания и программно-аппаратным ошибкам;
- ✓ обеспечение работы с файлами в сети.



## Типы файлов

**1. Обычные файлы** содержат информацию произвольного характера, которую заносит в них пользователь или программа, системная или пользовательская.

Содержание такого файла определяется приложением, которое с ним работает.



## Обычные файлы делятся на

### ✓ Файлы во внешнем представлении.

Условно можно назвать текстовыми. Они состоят из строк символов, представленных в ASCII-коде, и интерпретируются пользователем как текст в обычном понимании.

### ✓ Файлы во внутреннем представлении.

Условно можно назвать двоичными. Эти файлы создаются программным путем; их структура определяется программой – создателем.



**2. Специальные файлы** — это файлы, ассоциированные с устройствами ввода-вывода, которые позволяют пользователю выполнять операции ввода-вывода, используя обычные команды записи в файл или чтения из файла.



Тип	Значение	Примеры
<b>.doc, .txt</b>	Файл содержит текстовую информацию	MS Word, WordPad, Блокнот
<b>.bmp, .jpg, .gif</b>	Файл содержит графическую информацию	Adobe PhotoShop, ACDSee, Paint
<b>.avi</b>	Файл содержит видеoinформацию	WinAmp, Windows Медиа Проигрыватель
<b>.wav</b>	Файл содержит звуковую информацию	WinAmp, Windows Медиа Проигрыватель
<b>.bak</b>	Файл-копия	Открывается программой, в которой был создан файл
<b>.html, .htm</b>	Файл содержит Web-страницу	Internet Explorer
<b>.arj, .rar, .zip</b>	Файл содержит архив (сжатую информацию)	WinRar, WinZip
<b>.exe, .com</b>	Файл содержит программу – исполняемый файл	Открывает саму программу



## Атрибуты файлов

**Атрибут** – это информация, описывающая некоторое свойство файла, или некоторая характеристика файла.

- ✓ тип файла;
- ✓ создатель файла;
- ✓ пароль для доступа к файлу;
- ✓ информация о возможности доступа (права доступа);
- ✓ времена создания; последнего доступа и последнего изменения;
- ✓ текущий размер файла;
- ✓ признак “только для чтения”;
- ✓ признак “скрытый файл”;
- ✓ признак “архивный файл”.



## Дерево каталогов

В ОС Windows дисковое пространство разбивается пользователем на несколько логических дисководов, каждый из которых содержит свое дерево каталогов, не связанное с деревьями других дисководов.



**Каталог** — справочник файлов с указанием месторасположения на диске.

**Различают два состояния каталога:** текущее (активное) и пассивное.

**Текущий (активный) каталог** — каталог, в котором работа пользователя производится в текущее время.

**Пассивный каталог** — каталог, с которым в данный момент времени не имеется связи.



**Каталог высшего уровня** — главный, или  
корневой.

Он один на каждом диске, не имеет имени и  
обозначается в Windows знаком «\».



## **Простое, полное и относительное имя файла**

**Простое имя** идентифицирует файл в пределах каталога, в котором файл непосредственно размещен.

**Путь к файлу** – последовательность имен каталогов, разделенных знаком слеш (прямым или обратным, в зависимости от ОС), начиная с имени текущего каталога и до каталога, в котором находится файл.

**Полное имя файла в дереве каталогов** – путь от корня к файлу с приписанным через слеш простым именем файла.

**Относительное имя файла** – путь от текущего каталога к файлу с приписанным через слэш простым именем файла.



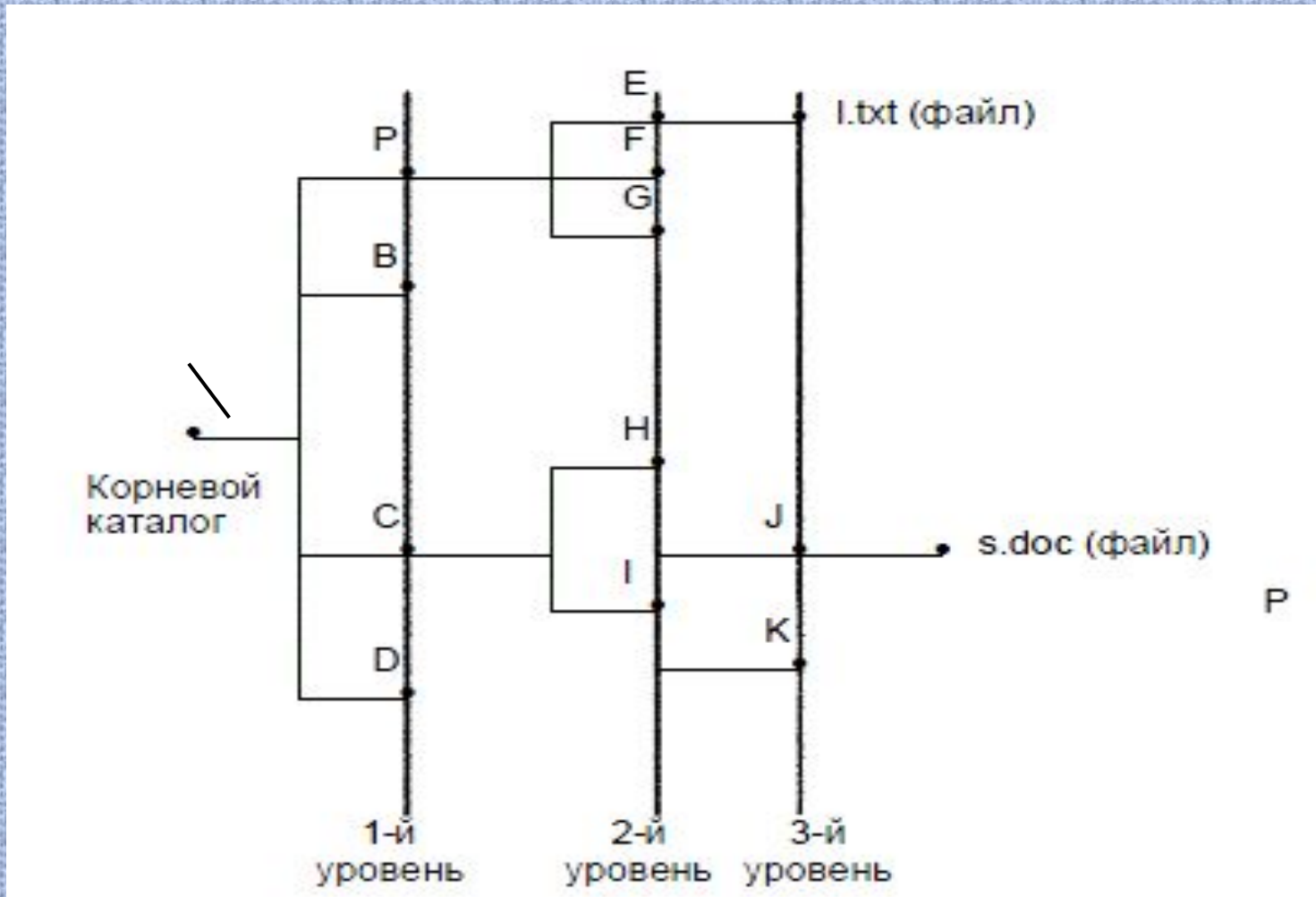
Если деревья расположены на разных логических дисках, то в полное имя в общем случае включается имя дисководов; в противном случае подразумевается текущий дисковод.

Полное имя однозначно идентифицирует файл.

Если путь не указан, то подразумевается текущий каталог.



# Пример дерева каталогов для Windows



**s.doc** – простое имя

**I\J\s.doc** – относительное имя

**C:\C\I\J\s.doc** – полное имя



# Логическая организация файлов

## Подходы к логической организации файлов

**Первый подход** предполагает, что единица данных для обмена с внешним устройством также осмысленна и определяется программистом.



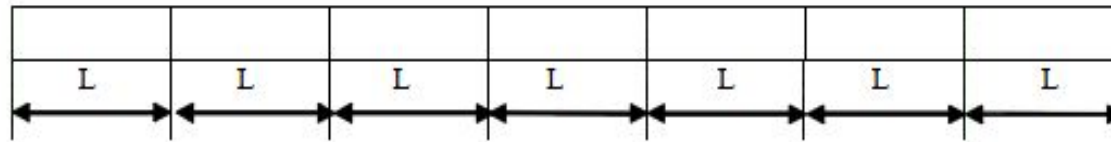
Способ доступа к записям файла определяет порядок их обработки (считывания – записи).

### **Возможны два способа доступа:**

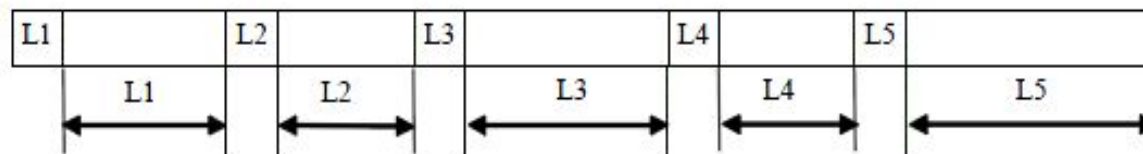
- 1. последовательный** – доступной для обработки является запись, непосредственно следующая за обработанной; так, если была обработана 3-я запись, то доступной является только 4-я; чтобы получить доступ к 5-й, надо обработать (хотя бы пропустить) 4-ю;
- 2. прямой** – каждая запись имеет некоторый ключ; доступной для обработки является запись с заданным ключом, вне зависимости от того, какая запись была доступна перед этим.



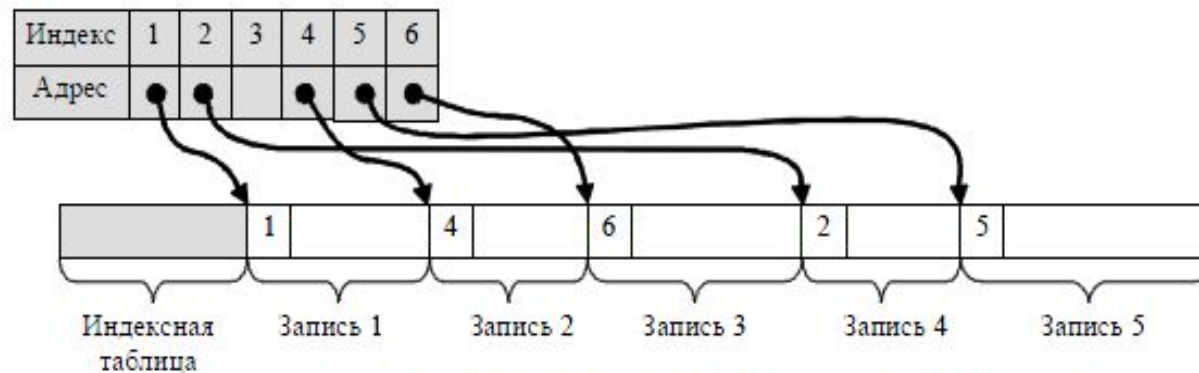
# Схемы структурирования



*Последовательная организация с записями фиксированной длины*



*Последовательная организация с записями переменной длины*



*Прямая организация. Индексированный файл*



**Второй подход** предполагает, что приложение полностью берет на себя интерпретацию содержимого файла, а обмен между оперативной и внешней памятью осуществляется последовательностями байтов заданной длины, начиная с заданной позиции.



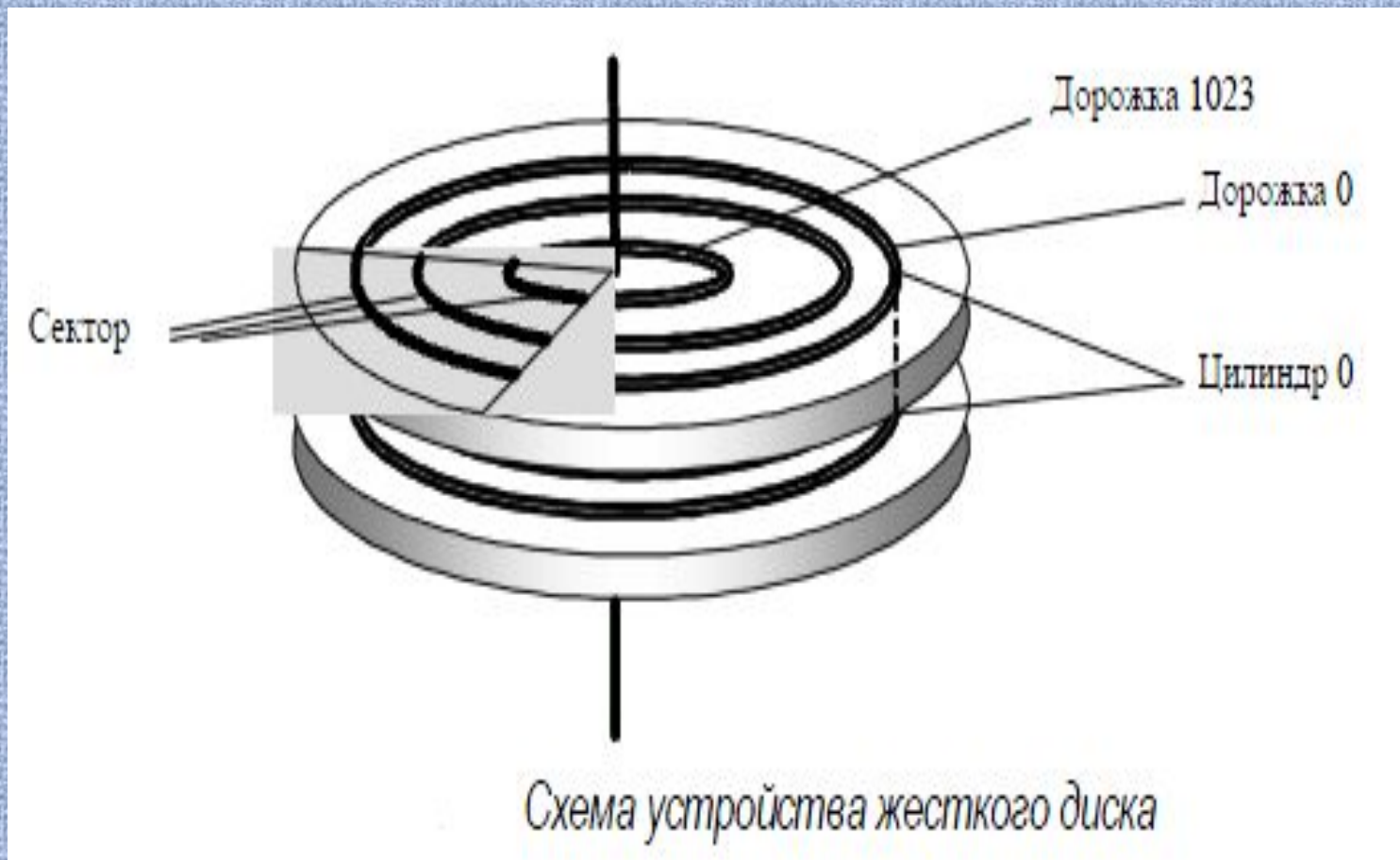
## Файловые операции

- ✓ открытие файла;
- ✓ закрытие файла;
- ✓ создание;
- ✓ чтение из файла;
- ✓ запись в файл;
- ✓ прямой доступ к файлу.

Эти операции представлены на уровне операционной системы в виде системных функций. Так, соответствующие функции в ОС Windows предоставляются пользовательским программным интерфейсом **WinAPI**.



# Физическая организация файловой системы





Жесткий диск в общем случае состоит из пакета **пластин**.

На каждой стороне каждой пластины размещены **дорожки** (tracks), на которых хранятся данные. Нумерация дорожек начинается с 0 от внешнего края диска.

Совокупность дорожек одного радиуса на всех поверхностях всех пластин называется **цилиндром**.

Каждая дорожка разбивается на **секторы**, или блоки фиксированного размера, кратного двум (как правило, 512 байтов). Все дорожки имеют одинаковое число секторов, поэтому плотность записи тем выше, чем ближе дорожка к центру.



Дорожки и секторы создаются при **физическом**, или **низкоуровневом форматировании** диска, предшествующем его использованию. Низкоуровневый формат не зависит от типа операционной системы, которая будет этот диск использовать.

**Сектор** — наименьшая физическая адресуемая единица обмена данными диска с оперативной памятью.



Доступ к секторам осуществляется на уровне контроллеров дисководов.

**Адрес сектора включает три составляющих:**

- ✓ номер цилиндра,
- ✓ номер поверхности,
- ✓ номер сектора.



Операционная система использует **логическую адресуемую единицу** дискового пространства — **кластер (cluster)**. При создании файла память на диске ему выделяется кластерами.



## Основные критерии эффективности физической организации файла:

- ✓ скорость доступа к данным;
- ✓ объем адресной информации файла;
- ✓ степень фрагментированности дискового пространства;
- ✓ возможность увеличения размера файла.



## Варианты физической организации

1. Непрерывное размещение.

2. Связанный список кластеров. В начале каждого кластера файла содержится указатель на следующий кластер. Расположение файла может быть задано номером первого кластера.



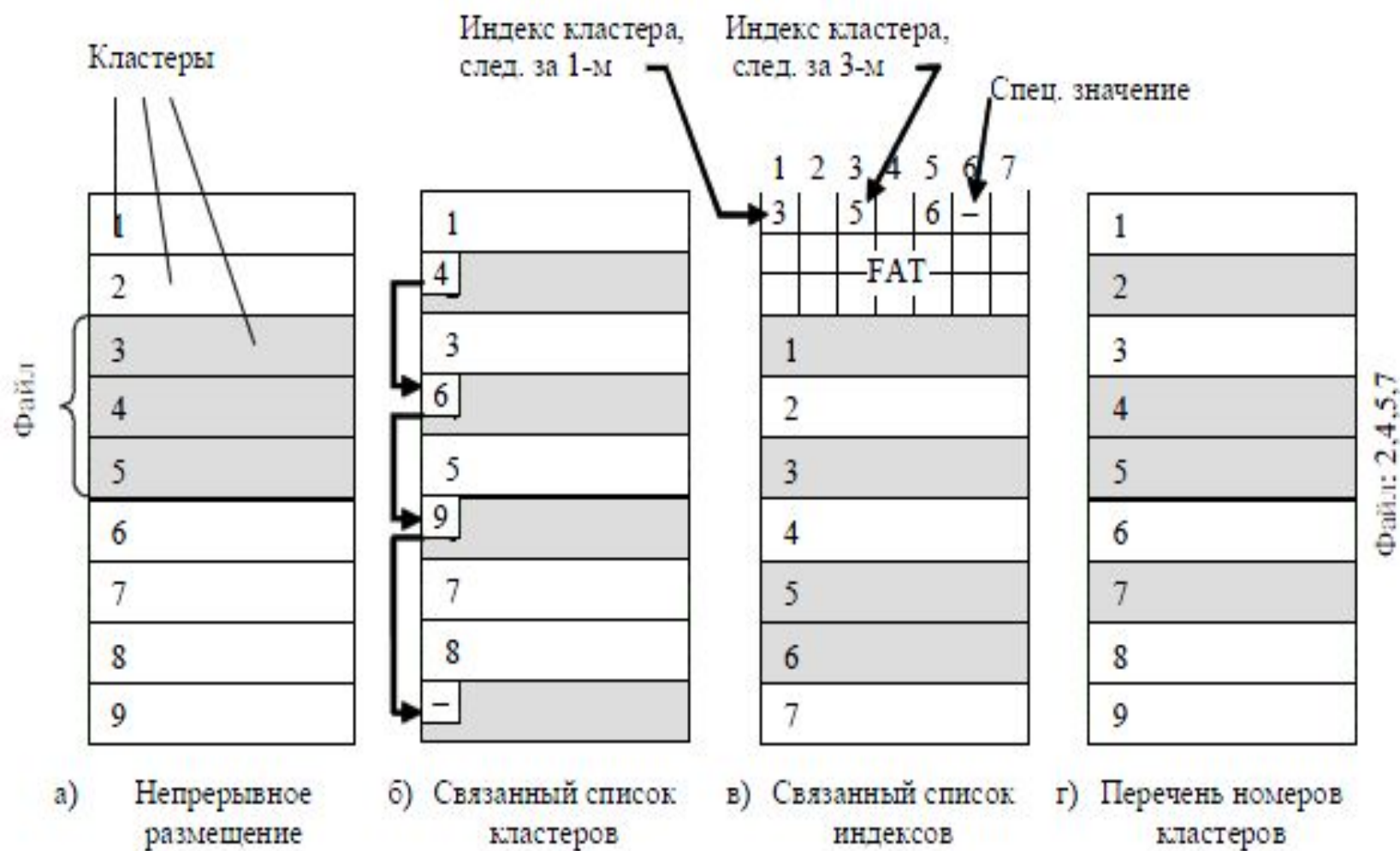
### 3. Связанный список индексов.

Применяется в файловой системе FAT для ОС семейства Microsoft.

С каждым кластером диска связывается индекс (номер). Индексы располагаются в отдельной таблице – FAT (File Allocation Table), занимающей один кластер. Когда память свободна, все индексы имеют нулевое значение.

4. Перечисление номеров кластеров, занимаемых файлом. Этот перечень служит адресом файла.





Физическая организация файла



# Сравнительная оценка вариантов

Критерий/ организация	Непрерывное размещение	Связанный список кластеров	Связанный список индексов	Перечисление номеров кластеров
<b>Скорость доступа</b>	высокая: нет затрат на поиск и считывание кластеров файла (+)	невысока: доступ к кластерам последовательны й (-)	высокая: доступ близок к прямому (+)	высокая: прямой доступ к кластеру (+)
<b>Объем адресной информации</b>	минимален: номер первого кластера и объем файла (+)	минимален: номер первого кластера (+)	минимален: номер первого кластера и FAT (+)	длина адреса зависит от размера файла (-)
<b>Степень фрагментаци и диска</b>	высокая (-)	на уровне кластеров фрагментация отсутствует (+)	на уровне кластеров фрагментация отсутствует (+)	на уровне кластеров фрагментация отсутствует (+)
<b>Возможность увеличения размера файла</b>	проблемы: необходимость выделения сплошного участка (-)	число кластеров легко наращивается (+)	число кластеров легко наращивается (+)	увеличение файла увеличивает длину адреса (-)